**Patrones para sistemas adaptables**

Stevenson morales

Sebastian palacio Tamayo

Andrés Loaiza Estrada

Asignatura:

Arquitectura de software

docente:

Paulo Andrés Arias Esguerra

Instituto Tecnológico Metropolitano

ITM

Septiembre de 2018

**Contenido**

[**1.** **Sistemas adaptables:** 3](#_Toc525920321)

[1.1. Causa y efecto 3](#_Toc525920322)

[1.2. Teoría de la complejidad 3](#_Toc525920323)

[1.3. Propiedades 3](#_Toc525920324)

[**2.** **Microkernel:** 5](#_Toc525920325)

[2.1. Contexto y Problema 5](#_Toc525920326)

[2.2. Solución, Consecuencias y Pasivos 5](#_Toc525920327)

[**3.** **Reflexión:** 7](#_Toc525920328)

[3.1. Estructura 7](#_Toc525920329)

[**Bibliografía** 10](#_Toc525920330)

# **Sistemas adaptables:**

## 1.1. Causa y efecto

Durante muchos años los científicos vieron el universo como un lugar lineal. Uno en el que se aplican simples reglas de causa y efecto. Ellos veían al universo como una máquina grande y pensaban que, si tomaban la máquina aparte y entendían las partes, entonces entenderían el todo.

También pensaban que los componentes del universo podían ser vistos como máquinas, creyendo que si trabajábamos en las partes de estas máquinas y hacíamos que cada parte funcionara mejor, entonces todo funcionaría mejor. Los científicos creían que el universo y todo lo que en él se podía predecir y controlar.

## 1.2. Teoría de la complejidad

Poco a poco, como los científicos de todas las disciplinas exploraron estos fenómenos una nueva teoría surgió la teoría de la complejidad, una teoría basada en las relaciones, la aparición, los patrones y las iteraciones. Una teoría que sostiene que el universo está lleno de sistemas, sistemas meteorológicos, sistemas inmunológicos, sistemas sociales, etc. y que estos sistemas son complejos y se adaptan constantemente a su entorno. De ahí los sistemas adaptativos complejos.

Los agentes en el sistema son todos los componentes de ese sistema. Por ejemplo, las moléculas de aire y agua en un sistema climático, y la flora y la fauna en un ecosistema. Estos agentes interactúan y se conectan entre sí de maneras impredecibles e imprevistas.

Pero a partir de esta masa de interacciones emergen las regularidades y empiezan a formar un patrón que se alimenta del sistema e informa las interacciones de los agentes. Por ejemplo, en un ecosistema, si un virus empieza a agotar una especie, esto resulta en un suministro mayor o menor de alimentos para otros en el sistema que afecta su comportamiento y su número. Se produce un período de flujo en todas las poblaciones del sistema hasta que se establece un nuevo equilibrio.

## 1.3. Propiedades

Los sistemas complejos adaptativos tienen muchas propiedades y los más importantes son:

* Aparición: En lugar de ser planificados o controlados, los agentes del sistema interactúan de manera aparentemente aleatoria. De todas estas interacciones emergen patrones que informan el comportamiento de los agentes dentro del sistema y el comportamiento del propio sistema. Por ejemplo, una colina de termitas es una pieza maravillosa de la arquitectura con un laberinto de pasajes de interconexión, grande cavernas, túneles de ventilación y mucho más. Sin embargo, no hay un gran plan, la colina sólo surge como resultado de las termitas siguiendo unas pocas reglas locales simples.
* Co-evolución: Todos los sistemas existen dentro de su propio entorno y también forman parte de ese entorno. Por lo tanto, como sus cambios de entorno que necesitan cambiar para garantizar el mejor ajuste. Pero porque son parte de su entorno, cuando cambian, cambian su entorno, y como ha cambiado necesitan cambiar de nuevo, y así continúa como un proceso constante. (Tal vez debería haber sido la "Teoría de la Co-evolución" de Darwin.). Algunas personas establecen una distinción entre sistemas adaptativos complejos y complejos sistemas evolutivos. Donde los primeros se adaptan continuamente a los cambios que los rodean, pero no aprenden del proceso. Y donde estos últimos aprenden y evolucionan de cada cambio que les permite influir en su entorno, predecir mejor los cambios probables en el futuro.
* Variedad requerida**:** Cuanto mayor sea la variedad dentro del sistema, más fuerte es. De hecho, la ambigüedad y la paradoja abundan en los complejos sistemas adaptativos que utilizan las contradicciones para crear nuevas posibilidades de co-evolucionar con su entorno. La democracia es un buen ejemplo en que su fuerza se deriva de su tolerancia e incluso insistencia en una variedad.
* Conectividad: Las formas en que los agentes de un sistema se conectan y se relacionan entre sí, es fundamental para la supervivencia del sistema, porque es a partir de estas conexiones que se forman los patrones y se difunde la retroalimentación. Las relaciones entre los agentes son generalmente más importantes que los propios agentes.
* Reglas simples: Los sistemas complejos adaptativos no son complicados. Los patrones emergentes pueden tener una variedad rica, pero como un caleidoscopio las reglas que gobiernan la función del sistema son bastante simples. Un ejemplo clásico es que todos los sistemas de agua en el mundo, todos los arroyos, ríos, lagos, océanos, cascadas, etc. con su infinita belleza, poder y variedad se rigen por el simple principio de que el agua encuentra su propio nivel.
* Iteración: Pequeños cambios en las condiciones iniciales del sistema pueden tener efectos significativos después de haber pasado por el bucle de emergencia-retroalimentación varias veces (a menudo denominado efecto mariposa). Una bola de nieve rodante, por ejemplo, ganancias en cada rollo mucha más nieve que lo hizo en el rollo anterior y muy pronto una bola de nieve tamaño puño.
* Auto organización: No hay jerarquía de mando y control en un complejo sistema adaptativo. No hay planificación o gestión, pero hay una constante reorganización para encontrar el mejor ajuste con el medio ambiente. Un ejemplo clásico es que si uno tomara cualquier ciudad occidental y agregara toda la comida en las tiendas y se dividiera por el número de personas en la ciudad habrá cerca de dos semanas suficiente suministro de alimentos, pero no hay plan de alimentos, Gerente de alimentos o cualquier otro proceso formal de control. El sistema se organiza continuamente a través del proceso de emergencia y retroalimentación.
* Borde del caos: La teoría de la complejidad no es lo mismo que la teoría del caos, que se deriva de la matemática. Pero el caos tiene un lugar en la teoría de la complejidad en que los sistemas existen en un espectro que va desde el equilibrio al caos. Un sistema en equilibrio no tiene la dinámica interna para permitirle responder a su entorno y morir lentamente (o rápidamente). Un sistema en el caos deja de funcionar como un sistema. El estado más productivo en el que se encuentra es en el borde del caos donde hay la máxima variedad y creatividad, dando lugar a nuevas posibilidades.
* Complejo adaptativo: los sistemas están a nuestro alrededor. La mayoría de las cosas que damos por sentadas son sistemas complejos adaptativos y los agentes de cada sistema existen y se comportan en total ignorancia del concepto pero eso no impide su contribución al sistema. Los sistemas complejos adaptativos son un modelo para pensar sobre el mundo que nos rodea no un modelo para predecir lo que sucederá.

# 

# **Microkernel:**

El patrón de Microkernel se aplica a los sistemas de software que deben ser capaces de adaptarse a los cambios en los requisitos del sistema. Separa un núcleo funcional mínimo de la funcionalidad extendida y de las piezas específicas del cliente. El microkernel también sirve como socket para conectar estas extensiones y coordinar su colaboración.

## 2.1. Contexto y Problema

El patrón puede aplicarse en el contexto de sistemas de software complejos que sirven de plataforma para otras aplicaciones de software. Tales sistemas complejos por lo general deben ser extensibles y adaptables a tecnologías emergentes, capaces de hacer frente a una gama de estándares y tecnologías. También necesitan poseer altas cualidades de rendimiento y escalabilidad; Como resultado, se requiere un bajo consumo de memoria y bajas demandas de procesamiento. En conjunto, los requisitos anteriores son difíciles de conseguir.

## 2.2. Solución, Consecuencias y Pasivos

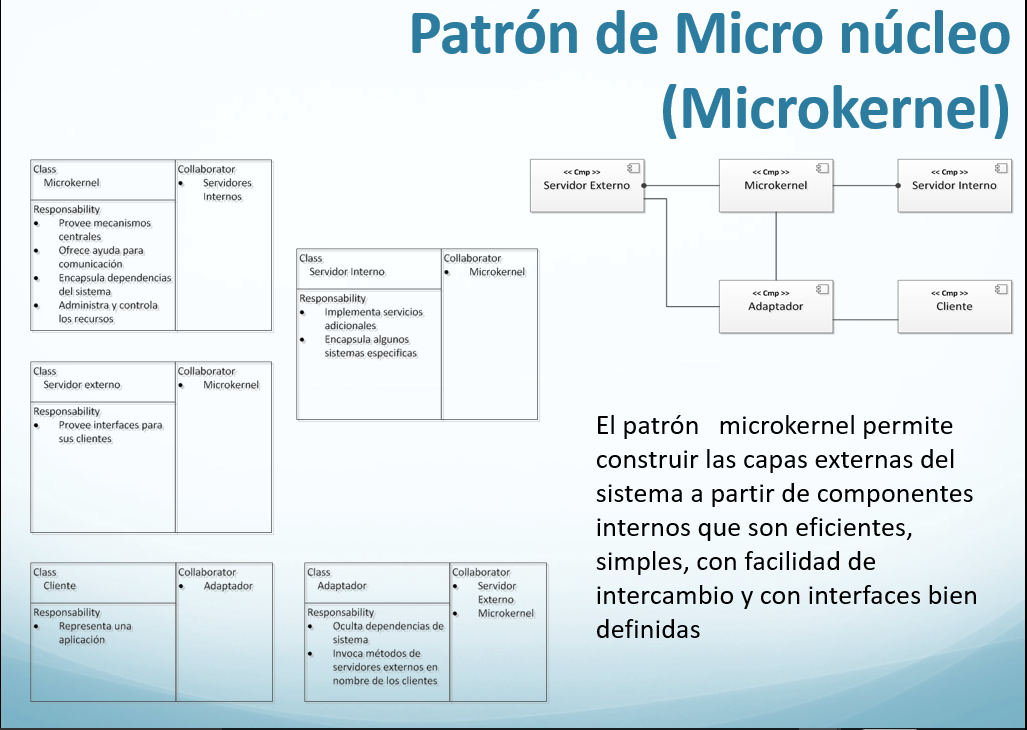
* Los servicios centrales más importantes del sistema deben estar encapsulados en un componente de microkernel. El microkernel mantiene los recursos del sistema y permite que otros componentes interactúen entre sí, así como para acceder a la funcionalidad del microkernel. Encapsula una parte significativa de las dependencias específicas del sistema, como los aspectos dependientes del hardware. El tamaño del microkernel debe mantenerse al mínimo, por lo tanto, solo se puede incluir parte de la funcionalidad del núcleo; El resto de la funcionalidad principal se difiere para separar los servidores internos.
* Los servidores internos amplían las funcionalidades del microkernel. Servidores internos puede por ejemplo manejar gráficos y medios de almacenamiento. Los servidores internos pueden tener sus propios procesos o se pueden compartir bibliotecas de vínculos dinámicos (DLL) cargadas dentro del kernel. El servidor externo proporciona una funcionalidad más compleja; Se construyen en la parte superior de los servicios básicos proporcionados por el microkernel.
* Diferentes servidores externos pueden ser necesarios en el sistema para proporcionar la funcionalidad para dominios de aplicación específicos. Estos servidores funcionan en procesos separados y emplean las facilidades de comunicación proporcionadas por el microkernel a las peticiones de los receptores de sus clientes y devuelven los resultados. El papel del adaptador es proporcionar una interfaz transparente para que los clientes se comuniquen con servidores externos. El adaptador oculta las dependencias del sistema, como las instalaciones de comunicación del cliente. El adaptador mejora así la escalabilidad y la capacidad de cambio del sistema. El adaptador permite distribuir los servidores y los clientes a través de una red.

Los beneficios del patrón se pueden mencionar como:

1. Buena portabilidad, ya que solo el microkernel necesita ser modificado al portar el sistema a un nuevo entorno.
2. Alta flexibilidad y extensibilidad, ya que las modificaciones o extensiones pueden realizarse modificando o ampliando servidores internos.
3. La separación de los mecanismos de bajo nivel (proporcionados por el microkernel y los servidores internos) y las políticas de nivel superior (proporcionadas por servidores externos) mejora la cambiabilidad del sistema.

También hay algunas preocupaciones.

1. El sistema de microkernel requiere mucha más comunicación entre procesos dentro de una ejecución de aplicación debido a las llamadas a servidores internos y externos.
2. El diseño e implementación del sistema basado en microkernel es mucho más complejo que el de un sistema monolítico



# **Reflexión:**

El patrón de reflexión proporciona un mecanismo para cambiar dinámicamente la estructura y el comportamiento de los sistemas de software. Apoya la modificación de aspectos fundamentales, tales como estructuras de tipo y mecanismos de llamada de función. En este patrón, una aplicación se divide en dos partes. Un meta nivel proporciona información sobre las propiedades del sistema seleccionadas y hace que el software sea consciente de sí mismo. Un nivel base incluye la lógica de la aplicación. Su implementación se basa en el nivel meta. Los cambios en la información guardada en el nivel meta afectan el comportamiento posterior de nivel básico.

El patrón arquitectónico de la Pizarra Reflectiva es independiente de los lenguajes de programación y los marcos de implementación específicos, y su uso puede minimizar la complejidad causada por la presencia de numerosas propiedades a nivel de sistema en masa.

## 3.1. Estructura

El subsistema de controlador está compuesto por un componente de protocolo de metaobjetos (MOP) que junto con una colección de meta-objetos implementan las estrategias de control de sistemas multi-agente. Los meta-objetos están compuestos de datos (metadatos) y son responsables de asociar comportamientos específicos (reacciones) a operaciones realizadas sobre partes específicas de datos.

Estos meta-objetos pueden modificar de forma transparente el comportamiento normal de la pizarra, implementando así las estrategias de control de sistemas multi-agente. Diferentes agentes pueden actuar sobre la pizarra por medio de sus sensores y realizadores, que pueden, respectivamente, detectar y realizar cambios en la pizarra que se puede considerar su entorno. Los agentes no se comunican directamente; Sólo escriben y leen datos de la pizarra.

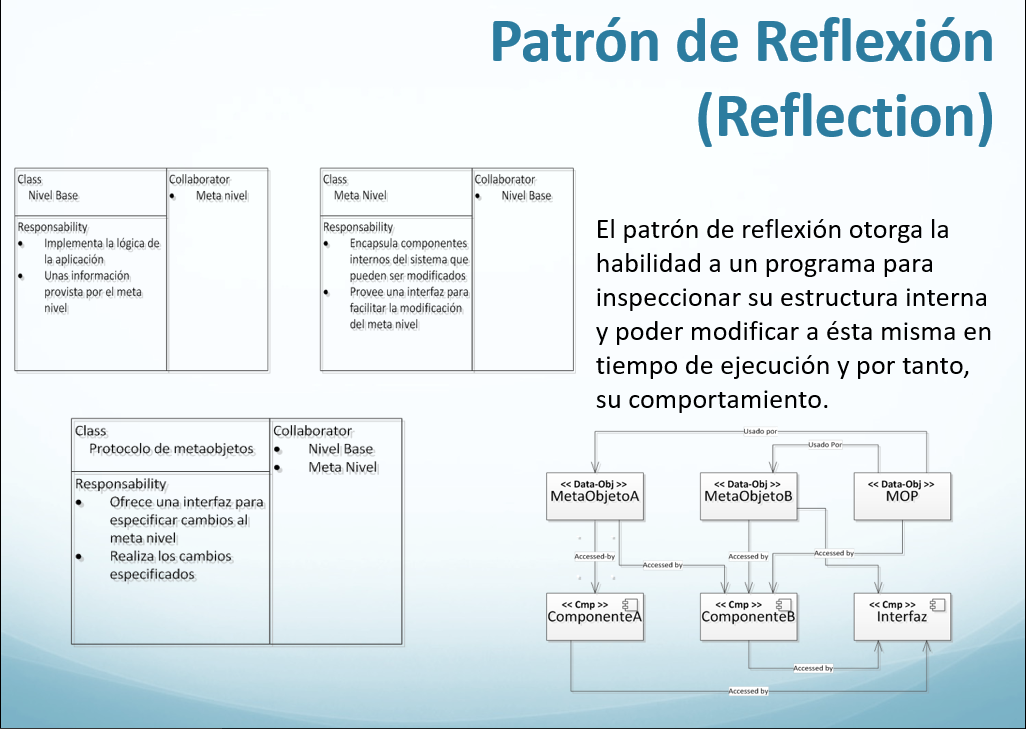
Siempre que un agente realice alguna operación sobre una pieza específica de datos almacenados en la pizarra, el componente MOP verifica si hay algún metaobjeto asociado a ella. Si es positiva, ejecuta la reacción asociada al metaobjeto, es decir, su comportamiento. La ejecución del metaobjeto puede acceder a la pizarra escribiendo y borrando datos.

De esta manera, en una arquitectura de pizarra reflexiva, la semántica de una operación de pizarra, de hecho, es el resultado de la ejecución de los metaobjetos asociados a ella. También pueden existir metaobjetos en el subsistema de control sin datos correspondientes en la pizarra. De esta manera, el sistema multi-agente puede asociar las reacciones a los datos que forman parte del vocabulario del sistema multi-agente y probablemente será escrito en la pizarra en tiempo de ejecución.

La reflexión se utiliza para interceptar y modificar los efectos de las operaciones de la pizarra.

Desde el punto de vista de los agentes de aplicación, la reflexión computacional es transparente: un agente escribe una pieza de datos en la pizarra y no tiene conocimiento de que esta operación de escritura ha sido interceptada y redirigida al meta-nivel. El siguiente escenario ilustra el comportamiento general de la arquitectura Reflective Blackboard:

1. Una fuente de conocimiento (o agente) realiza una operación en la pizarra (escribe por ejemplo), suministra una pieza de datos y espera alguna otra pieza de datos recuperados;
2. Esta operación es interceptada por el MOP del meta-nivel, que realizará, si se especifica, actividades de control sobre la operación realizada;
3. El MOP comprueba la existencia de metaobjetos asociados a los datos de pizarra y relacionados con la operación realizada. Si la fuente de conocimiento ha realizado una operación de escritura en la pizarra, los metaobjetos buscados serán aquellos relacionados con la pieza escrita de datos. Por otro lado, si la fuente de conocimiento ha realizado operaciones de lectura o eliminación, el metaobjeto buscado estará relacionado con la pieza de datos leída en la pizarra.



# **Bibliografía**

Información de patrones para sistemas adaptables en:

[https ://www.pdf-archive.com/2015/08/23/saunit6/preview/page/1/](https://www.pdf-archive.com/2015/08/23/saunit6/preview/page/1/)

23 de septiembre de 2017